AMENDMENTS

Please amend the above-referenced application as follows:

IN THE CLAIMS:

This listing of claims replaces all prior versions and listings of claims in the application.

1-23. (Canceled)

F1

24. (Currently Amended) A method for finding a predefined plurality of instructions, if available, that are ready to be executed and that reside in an instruction reordering mechanism of a processor that can launch execution of instructions out of order via a predefined number of instruction ports, comprising the steps of:

(a) providing said instruction reordering mechanism having a plurality of said instructions, each said instruction having a respective logic element for causing and preventing launching, when appropriate, of said instruction; and

(b) propagating a set of signals successively through said logic elements of said instruction reordering mechanism, said set of signals responsive to ~~available~~ instruction ports and port information, wherein an instruction identified as valid for launching is launched on a select instruction port identified via propagation logic forwarded through said logic elements and wherein said propagation logic includes a next available instruction port identifier when an instruction port is available.

25. (Previously Presented) The method of claim 24, further comprising the step of:

advising each instruction port of said instruction reordering mechanism during each launch cycle either that said instruction will be launched or that said instruction will not be launched.

26. (Previously Presented) The method of claim 24, wherein said signals are propagated through said logic elements only in response to logic transitions from a first logic level to a second logic level.

27. (Previously Presented) The method of claim 24, further comprising the step of:

communicating said predefined plurality of said instructions to a corresponding predefined plurality of instruction ports associated with one or more execution resources.

28. (Previously Presented) The method of claim 24, further comprising the step of, after said predefined plurality of said instructions have been selected, propagating a lost signal to remaining logic elements associated with remaining instructions of said instruction reordering mechanism to indicate to said remaining logic elements that their respective remaining instructions have not been selected.

29. (Previously Presented) The method of claim 24, further comprising the steps of:

(c) after said predefined plurality of said instructions have been selected, propagating a lost signal to remaining logic elements associated with remaining instructions of said instruction reordering mechanism to indicate to said remaining logic elements that their respective remaining instructions have not been selected;

(d) performing steps (b) and (c) during a single cycle associated with one or more execution resources; and

(e) communicating said predefined plurality of said instructions from said instruction reordering mechanism to a corresponding predefined plurality of instruction ports associated with said one or more execution resources.

30. (Previously Presented) The method of claim 24, further comprising the step of:

(c) providing said instruction reordering mechanism in a form of a queue having a plurality of slots, each said slot having a respective one of said logic elements and means for temporarily storing a respective instruction; and

(d) propagating said set of said signals successively through said slots of said queue during an execution cycle.

31. (Previously Presented) The method of claim 24, wherein said set comprises two or more signals.

32. (Previously Presented) The method of claim 24, further comprising the step of:

(c) causing said propagation through only a predefined number of said logic elements during a launch cycle.

P1

33. (Currently Amended) A method for quickly finding a predefined plurality of instructions, if available, that are ready to be executed and that reside in a queue of a processor that can launch the execution of instructions out of order, so that the found instructions can be communicated to a corresponding predefined plurality of ports associated with one or more execution resources, comprising the steps of:

(a) providing said queue having a plurality of slots, each said slot for temporarily storing a respective instruction and launching, when appropriate, execution of said respective instruction; and

(b) propagating a set of signals successively through slots of said queue during a launch cycle, said set of signals responsive to available instruction ports and port information to launch execution of an instruction that, when passed through a particular slot:

(1) select said particular slot for launching when said particular slot is ready by asserting in said slot one or more found signals that identify one or more specific ports associated with said one or more execution resources;

(2) refrain from selecting said particular slot when said particular slot is not ready by asserting in said slot a lost signal;

(3) track slots that have been selected during said launch cycle; and

(4) direct the selection of no more than said predefined plurality of said instructions during said launch cycle,

wherein an instruction identified as ready for launching is launched on a select instruction port identified via said set of signals forwarded through said queue, said set of signals comprising a next available instruction port identifier when an instruction port is available.

34. (Previously Presented) The method of claim 33, further comprising the step of:

communicating said predefined plurality of said instructions from said queue to said corresponding predefined plurality of ports associated with said one or more execution resources.

35. (Previously Presented) The method of claim 33, further comprising the step of:

(c) during said launch cycle but after said predefined plurality of said instructions have been selected, propagating a lost signal to remaining slots associated with remaining instructions of said queue to indicate to said remaining slots that their respective remaining instructions have not been selected.

36. (Currently Amended) A system for finding a predefined plurality of instructions, if available, that are ready to be executed in a processor that can launch execution of instructions out of order, comprising:

(a) an instruction reordering mechanism for temporarily storing a plurality of said instructions; and

(b) a plurality of logic elements associated with said instruction reordering mechanism and associated respectively with each of said instructions in said instruction reordering mechanism for causing and preventing launching, when appropriate, of respective instructions, said logic elements configured to propagate a plurality of signals through said logic elements, said plurality of signals responsive to available instruction ports and port information to launch execution of an instruction identified as valid for launching such that an instruction is launched on an instruction port identified via propagation logic forwarded through said logic elements said propagation logic comprising a next available instruction port identifier when an instruction port is available ~~that direct said logic elements to select said predefined plurality of said instructions for launching and de-select any remaining instructions~~.

37. (Previously Presented) The system of claim 36, wherein each of said logic elements is configured to receive said set of signals from a previous logic element, to evaluate said set of signals to determine whether or not to launch a respective instruction, to modify states associated with said set of signals based upon whether or not said respective instruction was launched, and to propagate said set of said signals to a later logic element.

38. (Previously Presented) The system of claim 36, wherein each one of said logic elements is implemented in combinational logic hardware.

39. (Previously Presented) The system of claim 36, wherein each said logic element is configured to, after said predefined plurality of said instructions have been selected, propagate a lost signal to remaining logic elements associated with said remaining instructions of said instruction reordering mechanism to indicate to said remaining logic elements that their respective remaining instructions have not been selected.

40. (Previously Presented) The system of claim 36, further comprising one or more execution resources having one or more ports to receive data from said predefined plurality of said instructions.

41. (Previously Presented) The system of claim 40, wherein at least one of said execution resources is an arithmetic logic unit (ALU).

42. (Previously Presented) The system of claim 40, wherein at least one of said execution resources is a multiple accumulate unit (MAC).

43. (Previously Presented) The system of claim 40, wherein at least one of said execution resources is a cache.

44. (Previously Presented) The system of claim 36, wherein said instruction reordering mechanism is a queue.

45. (Previously Presented) The system of claim 36, further comprising:

an arbitration mechanism configured to assert a start signal to one of said logic elements to initiate said propagation of said set of signals.

F1

46. (Currently Amended) A system for finding a predefined plurality of instructions, if available, that are ready to be executed and that reside in a queue of a processor that can launch execution of instructions out of order, comprising:

(a) queue means for storing a plurality of said instructions, said queue means having a plurality of launch logic means for causing and preventing launching, when appropriate, of a respective instruction; and

(b) logic means associated with said queue, said logic means for propagating a set of signals to successive launch logic means, said set of signals responsive to available instruction ports and port information to launch execution of an instruction to indicate both when and which of one or more ports of one or more execution resources are available for each said instruction and when none of said ports are available, wherein an instruction identified as valid for launching by said launch logic means is launched on an instruction port identified via said set of signals forwarded through said launch logic means and wherein said set of signals includes a next available instruction port identifier when an instruction port is available.